

# RECONSTRUCTING BIOCHEMICAL CLUSTER NETWORKS

UTZ-UWE HAUS, RAYMOND HEMMECKE, AND SEBASTIAN POKUTTA

**ABSTRACT.** Motivated by fundamental problems in chemistry and biology we study cluster graphs arising from a set of initial states  $S \subseteq \mathbb{Z}_+^n$  and a set of transitions/reactions  $M \subseteq \mathbb{Z}^n$ . The clusters are formed out of states that can be mutually transformed into each other by a sequence of reversible transitions. We provide a solution method from computational commutative algebra that allows for deciding whether two given states belong to the same cluster as well as for the reconstruction of the full cluster graph. Using the cluster graph approach we provide solutions to two fundamental questions: 1) Deciding whether two states are connected, e.g., if the initial state can be turned into the final state by a sequence of transition and 2) listing concisely all reactions processes that can accomplish that. As a computational example, we apply the framework to the permanganate/oxalic acid reaction.

## 1. INTRODUCTION

The reconstruction of all biochemical reaction networks composed of a given finite set of reactions/transitions that explain a given overall reaction from an initial state to some final state is one of the fundamental problems in biology and chemistry alike. The extraction of this decomposition is essential and has a wide range of applications from the design of large-scale reactors in process engineering where the presence of unexpected side products can disrupt the reaction process to the derivation of rate laws in physical chemistry.

One initial subproblem to be solved is to decide whether there exists a reaction network at all that explains the given overall reaction. For example, it has been settled only recently in [9], that if no additional catalyst is available, the 19 species postulated in (cf. [12]) do not suffice to explain the permanganate/oxalic acid reaction. So far, automatic tools to reconstruct the reaction networks are rare. Even the very promising recent approach using integer programming [11] suffered from ad-hoc assumptions. Apart from that, there have been considerable advances in the study of biochemical reaction networks using methods from discrete mathematics and computer algebra which are not directly related to our work although they use similar methods (see e.g., [7], [14], [17]).

We will formulate the underlying mathematical problem of computing cluster networks and present a solution approach from commutative algebra. Given two states, represented as vectors  $\mathbf{s}, \mathbf{t} \in \mathbb{Z}_+^n$  (these could be an initial and a final state of a biochemical reaction) and a set of potential transitions  $M = U \cup -U \cup D \subseteq \mathbb{Z}^n$  where  $U$  represents undirected transitions and  $D$  represents directed transitions, we say  $\mathbf{s}$  is  $M$ -connected to  $\mathbf{t}$  (short:  $\mathbf{s} \rightarrow_M \mathbf{t}$ ) if there exists a reaction path from  $\mathbf{s}$  to  $\mathbf{t}$  in  $M$ . We can formulate the following questions:

**Question 1.** Given two given states  $\mathbf{s}, \mathbf{t} \in \mathbb{Z}_+^n$ . Decide whether  $\mathbf{s} \rightarrow_M \mathbf{t}$ .

**Question 2.** Given two given states  $\mathbf{s}, \mathbf{t} \in \mathbb{Z}_+^n$  such that  $\mathbf{s} \rightarrow_M \mathbf{t}$ . Find all directed paths from  $\mathbf{s}$  to  $\mathbf{t}$  using only transitions in  $M$ .

For small instances, Question 1 and to some extent also Question 2 can be solved using a purely enumerative approach. This approach is finite if we assume that  $M$  is homogeneous with respect to

---

2000 *Mathematics Subject Classification.* 13P10, 05C20, 05C30, 05C40, 05C85, 92-08, 92C40, 92E10, 92E20.

*Key words and phrases.* Reaction mechanisms, computational chemistry, reactive intermediates, elementary reactions, reaction networks, chemical engineering, binomial ideals, gröbner bases, computer algebra.

some positive grading. This road was successfully pursued in [9] to examine the permanganate/oxalic acid reaction. Unfortunately, this approach fails for larger problem sizes due to the vast amount of possibilities that have to be enumerated as a consequence of the *combinatorial explosion*. The situation is especially challenging, as due to reversible reactions a large number of reaction paths are similar and thus block the view onto structurally different networks.

In this article, we provide an algorithmic solution to the following two problems. The first problem is the identification of equivalence classes (or *clusters*) of states in  $\mathbb{Z}_+^n$  induced by the reversible reactions  $U$ , that is, given  $\mathbf{u}, \mathbf{v} \in \mathbb{Z}_+^n$ , decide whether  $\mathbf{u} \leftrightarrow_U \mathbf{v}$ . The second problem is the construction of the graph of all clusters reachable from a given cluster. This cluster graph is the compressed reachability graph starting at  $\mathbf{s} \in \mathbb{Z}_+^n$  and using only transitions from  $M$ , where equivalent states with respect to  $U$  are contracted.

We will use techniques from computer algebra in order to transform the problems at hand into algebraic ones and then, using Gröbner bases, provide two algorithms that solve these problems. We will then propose a solution to Question 1 and Question 2 using the aforementioned cluster approach. The cluster graph that arises from contracting the state graph is usually considerably smaller and both questions can be decided on the cluster graph using traditional graph theoretic methods, thus enabling the successful processing of significantly larger instances. In particular, transitions connecting two clusters are of major importance for any decomposition. They can be easily identified within the cluster graph. Thus, being able to compute cluster graphs, one may study more complex reactions whose symmetries and equivalent paths block the view onto essential parts of the decomposition, see Figure 2.1.

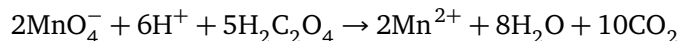
The outline of the article is as follows. As a motivation, we provide an introduction to the network reconstruction problem from a chemical point of view in Section 2. We will introduce the necessary notation and establish the link to our mathematical approach using directed graphs and commutative algebra. In Section 3 the necessary preliminaries and definitions from a mathematical point of view are formulated and basic results are established. We will then derive the described algorithms in Section 4 and provide computational results in Section 5. Finally, we conclude with some remarks in Section 6.

The notation we use is standard (cf. [4, 13]). These books also provide excellent introductions to commutative computer algebra.

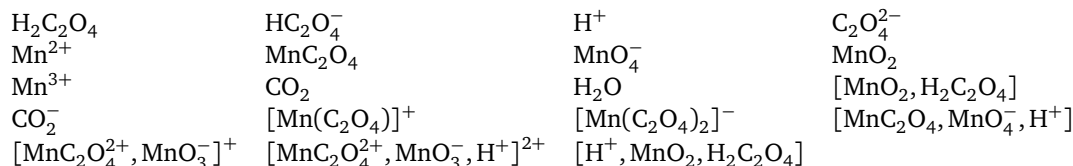
## 2. MOTIVATION FROM CHEMISTRY

The decomposition of an overall chemical reaction into elementary reaction steps is one of the fundamental questions in chemistry, since the network of elementary reactions encodes the dynamic of the chemical system and hence allows an analytical examination. We call a reaction *elementary* if at most two species react. For example,  $2\text{H}_2\text{O} \rightarrow 2\text{H}_2 + \text{O}_2$  is an elementary reaction, but the reverse reaction  $2\text{H}_2 + \text{O}_2 \rightarrow 2\text{H}_2\text{O}$  is not, as three (not necessarily different) species react.

The decomposition problem can be solve via several steps. We demonstrate them using the well-known permanganate/oxalic acid reaction



that has been studied since 1866 [8] and that has been studied by several groups (see e.g., [1], [2], [5], [12], [16], [18], [19]). As a first step, we have to choose a set of possible intermediate species that can be used to explain the overall reaction. For example, the following 19 species are a commonly accepted set [12]:



Now, all possible elementary reactions among the postulated species can be computed. For this, note that any chemical state can be represented by a vector in  $\mathbf{u} \in \mathbb{Z}_+^n$ , where  $u_i$  counts how many times species  $i$  is present in the state. Moreover, a chemical reaction can be encoded as an integer vector  $\mathbf{d} \in \mathbb{Z}^n$  (a difference of two states). In particular, in component  $d_i$  we specify how many units of species  $i$  react (if  $d_i < 0$ ) or how many units of species  $i$  are created (if  $d_i > 0$ ). As any chemical reaction must fulfill a balance of mass and charge,  $\mathbf{d}$  is a solution of a certain linear system of equations  $A\mathbf{z} = \mathbf{0}$  with  $A \in \mathbb{Z}^{m \times n}$  and  $\mathbf{z} \in \mathbb{Z}^n$ . For our example, we obtain

$$(2.1) \quad A = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 2 & 2 & 2 & 1 \\ 2 & 2 & 0 & 2 & 0 & 2 & 0 & 0 & 0 & 1 & 0 & 2 & 1 & 2 & 4 & 2 & 2 & 2 & 2 \\ 2 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 2 & 0 & 0 & 0 & 1 & 0 & 1 & 3 \\ 4 & 4 & 0 & 4 & 0 & 4 & 4 & 2 & 0 & 2 & 1 & 6 & 2 & 4 & 8 & 8 & 7 & 7 & 6 \\ 0 & 1 & -1 & 2 & -2 & 0 & 1 & 0 & -3 & 0 & 0 & 0 & 1 & -1 & 1 & 0 & -1 & -2 & -1 \end{pmatrix},$$

where the first four rows correspond to the mass balance equations of Mn, C, O, and H, respectively. The last row encodes balance of charge. The 19 columns correspond to the 19 postulated species. Thus, we naturally have  $a_i \geq 0$  for all  $i \in 1, \dots, m-1$ , motivated by the fact that each species contains at least one atom. The permanganate/oxalic acid reaction is encoded in the vector

$$\begin{pmatrix} -5 & 0 & -6 & 0 & 2 & 0 & -2 & 0 & 0 & 10 & 8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

The set of elementary reactions that we are looking for is now characterized by all integer vectors  $\mathbf{d} \in \ker(A)$  with  $\|\mathbf{d}^-\|_1 \leq 2$ . In our example, they can be computed via  $\binom{19}{2} + \binom{19}{1} + \binom{19}{1} = 209$  linear Diophantine systems. In total, they have 1022 solutions.

Finally, these 1022 reactions can be used to decompose the overall reaction into elementary steps. In particular, we are interested in all such possible reaction networks in order to find a network that explains the observations from experiments most consistently. From a computational point of view, this final decomposition step is by far the most challenging. For example, it was shown in [9] that the overall permanganate/oxalic acid reaction cannot be decomposed at all using the 1022 elementary reactions only. Thus, the 19 species do not suffice in order to explain this overall reaction. A solution was given by including a suitable additional species. Unfortunately, using this additional species and the resulting additional elementary reactions, the reaction network of the overall reaction contains a huge number of paths from the initial state  $2\text{MnO}_4^- + 6\text{H}^+ + 5\text{H}_2\text{C}_2\text{O}_4$  to the final state  $2\text{Mn}^{2+} + 8\text{H}_2\text{O} + 10\text{CO}_2$ . Many of these paths correspond to identical or essentially identical reaction networks. In order to identify important reactions within the network, we can cluster states together that are connected via a sequence of reversible reactions. Note that we may also cluster according to strongly connected components of states but clustering according to reversible reactions preserves more information of the original state network. In doing so, a large amount of the combinatorial explosion is removed. The resulting coarser network for the example at hand is given in Figure 2.1. This cluster network exhibits important elementary reactions needed to explain the overall reaction: For our example it shows that, given the postulated species, at most three essentially different reaction networks exist; some of those might turn out to be impossible due to chemical restrictions. Moreover, there is a unique reaction connecting cluster 0 to cluster 1, showing that this reaction must occur in *any* decomposition of the overall chemical reaction.

### 3. THE CLUSTER GRAPH FRAMEWORK

In this section, we want to introduce the necessary notions and definitions. We will also present a few fundamental results that we will use in the following exposition. The objects of our interest will be directed graphs and in particular their connectivity structure.

In the following, we will call the elements in  $\mathbb{Z}_+^n$  *states*. Let  $U \subseteq \mathbb{Z}^n$  be a set of *reversible (undirected) transitions*, let  $D \subseteq \mathbb{Z}^n$  be a set of *irreversible (directed) transitions*, and let  $M = U \cup -U \cup D$  be the set

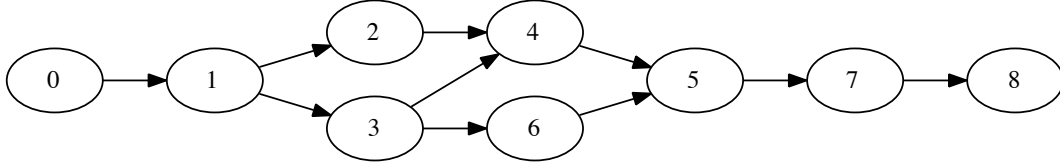


FIGURE 2.1. Cluster graph for permanganate/oxalic acid reaction. There is a unique reaction connecting cluster 0 to cluster 1, showing that it is essential in *any* decomposition.

of transitions. If  $\mathbf{s} \in \mathbb{Z}_+^n$  is a state and  $\mathbf{u} \in U$ , then we can transition to state  $\mathbf{s} - \mathbf{u}$  if  $\mathbf{s} - \mathbf{u} \geq 0$  or to state  $\mathbf{s} + \mathbf{u}$  if  $\mathbf{s} + \mathbf{u} \geq 0$  — in this sense the transitions in  $U$  are reversible. For the irreversible transitions in  $\mathbf{m} \in D$  we can (only!) transition to  $\mathbf{s} + \mathbf{m}$  if  $\mathbf{s} + \mathbf{m} \geq 0$ . Thus, we construct the infinite graph  $\Gamma_M$  with node set  $\mathbb{Z}_+^n$  and with an arc from  $\mathbf{a} \in \mathbb{Z}_+^n$  to  $\mathbf{b} \in \mathbb{Z}_+^n$  if and only if  $\mathbf{b} - \mathbf{a} \in M$ . Given two states  $\mathbf{s}, \mathbf{t} \in \mathbb{Z}_+^n$ , we say that  $\mathbf{s}$  is  $M$ -connected to  $\mathbf{t}$  (short:  $\mathbf{s} \rightarrow_M \mathbf{t}$ ) if there exists a directed path in  $\Gamma_M$  from  $\mathbf{s}$  to  $\mathbf{t}$ .

Moreover, we assume that  $M$  is homogeneous with respect to a positive (multi-) grading  $\deg : \mathbb{Z}_+^n \rightarrow \mathbb{Z}_+^k$ . This implies that  $M \cap \mathbb{Z}_+^n = \emptyset$  and that there are only finitely many states with a given fixed degree. In particular, there are only finitely many states reachable from any given state using the transitions from  $M$ . We outline this in more detail in the following example.

**Example 3.1.** Consider the matrix  $A \in \mathbb{Z}^{5 \times 19}$  given in (2.1) which is the mass/charge balance matrix for the permanganate/oxalic acid reaction assuming 19 species.

Adding up the first 4 rows we obtain the vector:

$$a_1 + a_2 + a_3 + a_4 = \begin{pmatrix} 8 & 7 & 1 & 6 & 1 & 7 & 5 & 3 & 1 & 3 & 3 & 11 & 3 & 7 & 13 & 13 & 11 & 12 & 12 \end{pmatrix},$$

which gives a desired positive grading. We can in fact also obtain a positive multi-grading by adding the vector  $a_1 + a_2 + a_3 + a_4$  sufficiently often to the rows of  $A$ :

$$\bar{A} = \begin{pmatrix} 8 & 7 & 1 & 6 & 1 & 8 & 6 & 4 & 2 & 3 & 3 & 12 & 3 & 8 & 14 & 15 & 13 & 14 & 13 \\ 10 & 9 & 1 & 8 & 1 & 9 & 5 & 3 & 1 & 4 & 3 & 13 & 4 & 9 & 17 & 15 & 11 & 14 & 14 \\ 10 & 8 & 2 & 6 & 1 & 7 & 5 & 3 & 1 & 3 & 5 & 13 & 3 & 7 & 13 & 14 & 11 & 13 & 15 \\ 12 & 11 & 1 & 10 & 1 & 11 & 9 & 5 & 1 & 5 & 4 & 17 & 5 & 11 & 21 & 21 & 18 & 19 & 18 \\ 32 & 29 & 3 & 22 & 2 & 28 & 19 & 12 & 1 & 12 & 12 & 44 & 13 & 27 & 53 & 52 & 43 & 46 & 47 \end{pmatrix}$$

Then the multi-grading is obtained by setting  $\deg(\mathbf{u}) := \bar{A}\mathbf{u}$ .

Now let us partition the graph  $\Gamma_M$  into clusters. This approach was initially suggested in [9]. However, in contrast to partitioning via the strongly connected components of  $\Gamma_M$  as in [9], we partition via equivalence classes which group all states  $\mathbf{v}$  that can be reached from a specific  $\mathbf{u}$  using only transitions from  $U$ . These equivalence classes will be called *clusters*. Note, that  $\mathbf{u} \rightarrow_U \mathbf{v}$  if and only if  $\mathbf{v} \rightarrow_U \mathbf{u}$ . Thus the relation  $\mathbf{u} \rightarrow_U \mathbf{v}$  is reflexive, symmetric, and transitive and therefore indeed an equivalence relation on  $\Gamma_M$  and we write  $\mathbf{u} \leftrightarrow_U \mathbf{v}$ . By  $\mathcal{C}(\mathbf{u}) \subseteq \mathbb{Z}_+^n$  we denote the cluster of  $\mathbf{u} \in \mathbb{Z}_+^n$  which is defined in the canonical way, i.e.,

$$\mathcal{C}(\mathbf{u}) := \{\mathbf{v} \in \mathbb{Z}_+^n \mid \mathbf{u} \leftrightarrow_U \mathbf{v}\}.$$

Now, the remaining set  $D \subseteq \mathbb{Z}^n$  of irreversible transitions defines a directed graph with the clusters (of  $\mathbb{Z}_+^n$ ) as vertices and with a directed edge from cluster  $\mathcal{C}(\mathbf{u})$  to cluster  $\mathcal{C}(\mathbf{v})$  if there exists two states  $\mathbf{x} \in \mathcal{C}(\mathbf{u})$  and  $\mathbf{y} \in \mathcal{C}(\mathbf{v})$  such that  $\mathbf{y} - \mathbf{x} \in D$ . We denote this cluster graph by  $\mathcal{G}(U, D)$ . Be aware that

we indeed only add the arc  $(\mathcal{C}(\mathbf{u}), \mathcal{C}(\mathbf{v}))$  if the cluster  $\mathcal{C}(\mathbf{v})$  can be reached from the cluster  $\mathcal{C}(\mathbf{u})$  by a transition path of length 1.

In applications, we are usually given a finite set  $S$  of initial states. Note that in our chemical setting,  $S$  typically contains only one state. More than one initial state could occur if a set of experiments (i.e., pairs of initial and final states) have to be explained (by networks). The nodes of the subgraph  $\bar{\Gamma} = (\bar{S}, \bar{M})$  of  $\Gamma_M$  reachable from  $S$  in  $\Gamma_M$  are

$$\bar{S} = \left\{ \mathbf{s} + \sum_{i=1}^k \mathbf{v}_i \mid k \in \mathbb{N}, \mathbf{s} \in S, (\mathbf{v}_i)_{i \in [k]} \subseteq M \text{ s.t. } \mathbf{s} + \sum_{i=1}^l \mathbf{v}_i \geq 0 \ \forall l \in [k] \right\}.$$

Herein, we use for convenience  $[k] := \{1, \dots, k\}$  for  $k \in \mathbb{N}$ . As the arc set  $\bar{M} \subseteq \bar{S} \times \bar{S}$  we obtain the set of those arcs in  $\Gamma_M$  involving only nodes from  $\bar{S}$ . The important difference to a classical directed graph is now that  $\bar{M}$  and  $\bar{S}$  are not given explicitly but implicitly by a set of transitions/reactions  $M = U \cup -U \cup D \subseteq \mathbb{Z}^n \setminus \mathbb{Z}_+^n$  and a set of initial states  $S$ . We call  $\bar{\Gamma} = (\bar{S}, \bar{M})$  the *state graph* of  $S$  and with respect to  $M$ . As  $\bar{S}$  and  $\bar{M}$  might be already very large for small instances it is favorable to not completely calculate  $\bar{S}$  and  $\bar{M}$ . In these cases, computing the much smaller cluster graph may still give important information on the decomposition of the overall reaction.

As  $M$  is assumed to be homogeneous with respect to some (positive) grading, the state graph of  $S$  w.r.t.  $M$  decomposes for  $\mathbf{s}_1, \mathbf{s}_2$  with different degrees and thus one can confine the analysis to sets  $S$  with elements of the same degree.

Having provided the considered setting, we set out to provide algorithmic solutions to answer Questions 1 and 2 using cluster graphs. For this we show how to construct the part  $\mathcal{G}(U, D, S)$  of the cluster graph  $\mathcal{G}(U, D)$  reachable from  $\mathcal{C}(\mathbf{s})$  with  $\mathbf{s} \in S$ .

#### 4. RECONSTRUCTING CLUSTER GRAPHS

In this section we will use computer algebraic tools to reconstruct the cluster graph reachable from states given in a finite set  $S$  via transitions in  $M = U \cup -U \cup D$ . The positive grading implies that each cluster contains only finitely many states. Moreover, although the total cluster graph over  $\mathbb{Z}_+^n$  has infinitely many clusters as vertices, the positive grading implies that for any given state  $\mathbf{s} \in \mathbb{Z}_+^n$  only finitely many clusters can be reached from the clusters  $\mathcal{C}(\mathbf{s})$  with  $\mathbf{s} \in S$ , within the cluster graph. In order to reconstruct  $\mathcal{G}(U, D, S)$ , we have to solve the following two problems:

**Main Problem 1.** Given two states  $\mathbf{s}, \mathbf{t} \in \mathbb{Z}_+^n$ . Decide whether  $\mathcal{C}(\mathbf{s}) = \mathcal{C}(\mathbf{t})$ , i.e., whether  $\mathbf{s} \leftrightarrow_U \mathbf{t}$ .

**Main Problem 2.** Given  $\mathbf{s} \in \mathbb{Z}_+^n$ . List all transitions  $\mathbf{d} \in D$  that are applicable to at least one state  $\mathbf{s}_d \in \mathcal{C}(\mathbf{s})$ , that is,  $\mathbf{s}_d + \mathbf{d} \geq 0$ .

Main Problem 1 captures the problem of being able to decide whether two states are in the same equivalence class whereas Main Problem 2 has to be solved in order to identify clusters reachable from a given cluster. We will provide a solution for both problems in form of computer algebraic algorithms. If the clusters are small enough such that all states can be enumerated explicitly, Main Problem 1 can be solved by simple enumeration. However, the sizes of the clusters may prohibit an explicit enumeration. Therefore an approach is sought that does not explicitly enumerate the cluster elements. We will tackle this problem by transforming the set  $U$  into a binomial ideal. We consider the polynomial ring  $K[X]$  where  $K$  is an arbitrary field and  $X$  is the set of the variables. Let  $J_U$  be defined to be the ideal

$$J_U := \langle \mathbf{x}^{\mathbf{u}^+} - \mathbf{x}^{\mathbf{u}^-} : \mathbf{u} \in U \rangle \subseteq K[X].$$

Then  $\mathbf{y} \leftrightarrow_U \mathbf{z}$  if and only if  $\mathbf{x}^{\mathbf{y}} - \mathbf{x}^{\mathbf{z}} \in J_U$  as shown in the following theorem. This theorem has been rediscovered many times — an early reference is [15]. For the sake of completeness we include a proof below (cf. [10]).

**Theorem 4.1.** Let  $U \subseteq \mathbb{Z}^n$  and  $\mathbf{y}, \mathbf{z} \in \mathbb{Z}_+^n$ . Then  $\mathbf{y} \leftrightarrow_U \mathbf{z}$  if and only if  $\mathbf{x}^{\mathbf{y}} - \mathbf{x}^{\mathbf{z}} \in J_U$ .

*Proof.* Let  $\mathbf{y}, \mathbf{z} \in \mathbb{Z}_+^n$  such that  $\mathbf{y} \leftrightarrow_U \mathbf{z}$ . Thus, there exists a path from  $\mathbf{y}$  to  $\mathbf{z}$  in  $\mathbb{Z}_+^n$ . More specifically, there exists a sequence of points  $(\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_k) \subseteq \mathbb{Z}_+^n$  where  $\mathbf{p}_1 = \mathbf{y}$ ,  $\mathbf{p}_k = \mathbf{z}$ , and  $\mathbf{p}_i - \mathbf{p}_{i+1} = \delta_i \mathbf{u}_i$  for some  $\mathbf{u}_i \in U$  and  $\delta_i \in \{1, -1\}$  for  $i = 1, \dots, k-1$ . Thus, there exists  $\boldsymbol{\gamma}_i \in \mathbb{N}^n$  such that  $\mathbf{p}_i = (\delta_i \mathbf{u}_i)^+ + \boldsymbol{\gamma}_i$ , and  $\mathbf{p}_{i+1} = (\delta_i \mathbf{u}_i)^- + \boldsymbol{\gamma}_i$  for every  $i = 1, \dots, k-1$ . Hence,  $\mathbf{x}^{\mathbf{p}_i} - \mathbf{x}^{\mathbf{p}_{i+1}} = \delta_i \mathbf{x}^{\boldsymbol{\gamma}_i} (\mathbf{x}^{\mathbf{u}_i^+} - \mathbf{x}^{\mathbf{u}_i^-})$ , and therefore,

$$\mathbf{x}^{\mathbf{y}} - \mathbf{x}^{\mathbf{z}} = \sum_{i=1}^{k-1} \mathbf{x}^{\mathbf{p}_i} - \mathbf{x}^{\mathbf{p}_{i+1}} = \sum_{i=1}^{k-1} \delta_i \mathbf{x}^{\boldsymbol{\gamma}_i} (\mathbf{x}^{\mathbf{u}_i^+} - \mathbf{x}^{\mathbf{u}_i^-}) \in J_U$$

as required.

Conversely, assume that  $\mathbf{x}^{\mathbf{y}} - \mathbf{x}^{\mathbf{z}} \in J_U$ . Further, suppose for contradiction,  $\mathbf{y} \not\leftrightarrow_U \mathbf{z}$ . As  $\mathbf{x}^{\mathbf{y}} - \mathbf{x}^{\mathbf{z}} \in J_U$ , we may write  $\mathbf{x}^{\mathbf{y}} - \mathbf{x}^{\mathbf{z}} = \sum_{i=1}^d c_i \mathbf{x}^{\boldsymbol{\gamma}_i} (\mathbf{x}^{\mathbf{u}_i^+} - \mathbf{x}^{\mathbf{u}_i^-})$  where  $\mathbf{u}_i \in U$ ,  $c_i \in K$ , and  $\boldsymbol{\gamma}_i \in \mathbb{N}^n$ . Note that we allow  $\mathbf{u}_i = \mathbf{u}_j$  for  $i \neq j$ . Now, let  $I := \{i \in [d] \mid (\boldsymbol{\gamma}_i + \mathbf{u}_i^+) \leftrightarrow_U \mathbf{y}\}$ . Clearly,  $(\boldsymbol{\gamma}_i + \mathbf{u}_i^+) \in \mathbb{Z}_+^n$  for all  $i \in [d]$ . Note that if  $(\boldsymbol{\gamma}_i + \mathbf{u}_i^+) \leftrightarrow_U \mathbf{y}$  then  $(\boldsymbol{\gamma}_i + \mathbf{u}_i^-) \leftrightarrow_U \mathbf{y}$  since  $(\boldsymbol{\gamma}_i + \mathbf{u}_i^+) - (\boldsymbol{\gamma}_i + \mathbf{u}_i^-) = \mathbf{u}_i \in U$ . Thus, the set of monomials consisting of  $\mathbf{x}^{\boldsymbol{\gamma}_i} \mathbf{x}^{\mathbf{u}_i^+}$  and  $\mathbf{x}^{\boldsymbol{\gamma}_i} \mathbf{x}^{\mathbf{u}_i^-}$  for all  $i \in I$ , which includes  $\mathbf{x}^{\mathbf{y}}$  and not  $\mathbf{x}^{\mathbf{z}}$ , is disjoint from the set of monomials consisting of  $\mathbf{x}^{\boldsymbol{\gamma}_i} \mathbf{x}^{\mathbf{u}_i^+}$  and  $\mathbf{x}^{\boldsymbol{\gamma}_i} \mathbf{x}^{\mathbf{u}_i^-}$  for all  $i \notin I$ , which includes  $\mathbf{x}^{\mathbf{z}}$  and not  $\mathbf{x}^{\mathbf{y}}$ . Let  $f(\mathbf{x}) = \sum_{i \in I} c_i \mathbf{x}^{\boldsymbol{\gamma}_i} (\mathbf{x}^{\mathbf{u}_i^+} - \mathbf{x}^{\mathbf{u}_i^-})$  and let  $g(\mathbf{x}) = -\sum_{i \notin I} c_i \mathbf{x}^{\boldsymbol{\gamma}_i} (\mathbf{x}^{\mathbf{u}_i^+} - \mathbf{x}^{\mathbf{u}_i^-})$ . It is readily seen that the polynomials  $f(\mathbf{x})$  and  $g(\mathbf{x})$  have a disjoint set of monomials, and therefore,  $f(\mathbf{x}) = \mathbf{x}^{\mathbf{y}}$  and  $g(\mathbf{x}) = \mathbf{x}^{\mathbf{z}}$  since  $\mathbf{x}^{\mathbf{y}} - \mathbf{x}^{\mathbf{z}} = f(\mathbf{x}) - g(\mathbf{x})$ . However, this is impossible since  $f(\mathbf{1}) = 0$  and  $g(\mathbf{1}) = 0$  but  $\mathbf{1}^{\mathbf{y}} = 1$  and  $\mathbf{1}^{\mathbf{z}} = 1$ .  $\square$

Using Theorem 4.1 we can now solve Main Problem 1 as follows. Choose  $\prec$  to be an arbitrary term ordering, let  $G_{\prec}(J_U)$  be a Gröbner basis of  $J_U$  with respect to  $\prec$ , and let  $\mathbf{y}, \mathbf{z} \in \mathbb{Z}_+^n$  be two states. Then  $\mathbf{y} \leftrightarrow_U \mathbf{z}$  if and only if  $\mathbf{x}^{\mathbf{y}} - \mathbf{x}^{\mathbf{z}} \in J_U$  by Theorem 4.1. As  $G_{\prec}(J_U)$  is a Gröbner basis,  $\mathbf{x}^{\mathbf{y}} - \mathbf{x}^{\mathbf{z}} \in J_U$  if and only if  $\text{NF}_{\prec}(\mathbf{x}^{\mathbf{y}} - \mathbf{x}^{\mathbf{z}}, G) = 0$ , where  $\text{NF}_{\prec}(\cdot)$  is the normal form operator. Note that once the Gröbner basis  $G_{\prec}(J_U)$  of  $J_U$  has been calculated, the membership test  $\text{NF}_{\prec}(\mathbf{x}^{\mathbf{y}} - \mathbf{x}^{\mathbf{z}}, G) = 0$  can be performed easily.

As an immediate consequence of this construction, it follows that every cluster  $\mathcal{C}(\mathbf{s})$  with  $\mathbf{s} \in \mathbb{Z}_+^n$  has a unique ( $\prec$ -minimal) representative outside the leading term ideal  $\text{LT}_{\prec}(J_U)$  which can be obtained by calculating the normal form. Slightly abusing notation we denote this unique minimal element by  $\mathcal{C}(\mathbf{s}) := \text{NF}_{\prec}(\mathbf{x}^{\mathbf{s}}, G)$ . Thus, the monomials outside of  $\text{LT}_{\prec}(J_U)$  are in one-to-one correspondence with possible clusters in  $\mathbb{Z}_+^n$ . Consequently, the number of clusters reachable from  $\mathcal{C}(\mathbf{s})$  is bounded by the number of monomials outside of  $\text{LT}_{\prec}(J_U)$  which have the same (multi-) degree  $\deg(\mathbf{s})$ . These numbers, however, are encoded in the multi-graded Hilbert-Poincaré series of  $J_U$ :

$$\mathcal{H}_{(\mathbf{z})} := \mathcal{H}(\mathbf{z}, J_U, \prec, \deg) := \sum_{\boldsymbol{\alpha} \in \mathbb{Z}_+^n : \mathbf{x}^{\boldsymbol{\alpha}} \notin \text{LT}_{\prec}(J_U)} \mathbf{z}^{\deg(\boldsymbol{\alpha})}.$$

Note that this potentially infinite sum can always be written as a rational function  $p(\mathbf{z})/q(\mathbf{z})$ . Moreover, it does not depend on the actual term ordering  $\prec$  chosen and it can be computed from any Gröbner basis of  $J_U$  rather efficiently. The coefficient in front of  $\mathbf{z}^{\deg(\mathbf{s})}$  in the multi-variate Hilbert series expansion of  $p(\mathbf{z})/q(\mathbf{z})$  is exactly the number of clusters of degree  $\deg(\mathbf{s}) \in \mathbb{Z}^k$ . Thus, we can compute in advance an upper bound on the number of clusters that can be reached from  $\mathcal{C}(\mathbf{s})$  or that can reach  $\mathcal{C}(\mathbf{s})$  inside the cluster graph  $\mathcal{G}(U, D)$ .

Recall that for a directed transition  $\mathbf{d} \in D$  and a state  $\mathbf{z} \in \mathbb{Z}_+^n$  the transition  $\mathbf{d}$  is applicable to  $\mathbf{z}$ , if  $\mathbf{z} + \mathbf{d} \in \mathbb{Z}_+^n$ . Thus, in order to solve Main Problem 2, we need to decide whether a given directed transition  $\mathbf{d} \in \mathbb{Z}^n$  is applicable to some state  $\mathbf{z} \in \mathcal{C}(\mathbf{y})$  given only some representative  $\mathbf{y}$  of the cluster  $\mathcal{C}(\mathbf{y})$ . The following theorem will be a main ingredient in order to solve this question. Recall that for an ideal  $I \subseteq K[X]$  and a vector  $\mathbf{m} \in \mathbb{Z}^n$  the colon ideal  $I : \mathbf{x}^{\mathbf{m}}$  is defined as  $I : \mathbf{x}^{\mathbf{m}} := \{p \in K[X] \mid p\mathbf{x}^{\mathbf{m}} \in I\}$ .

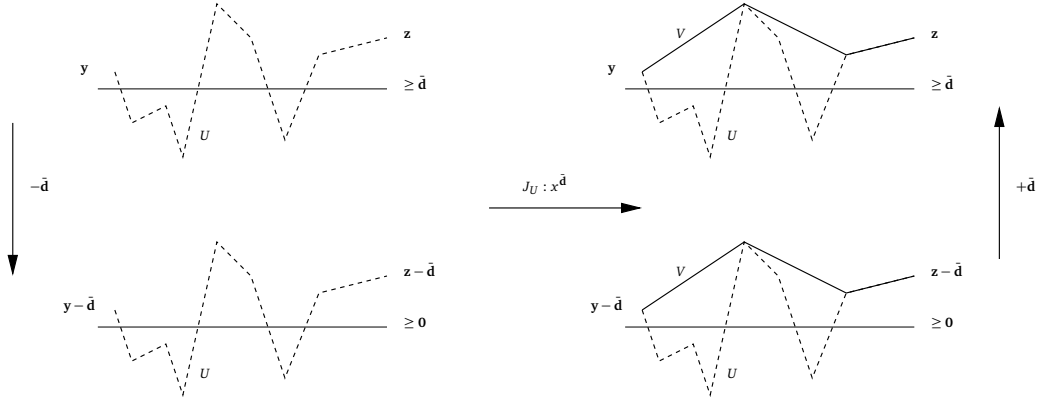


FIGURE 4.1. Shift relative to  $\bar{\mathbf{d}}$ .

**Theorem 4.2.** Let  $U \subseteq \mathbb{Z}^n$ ,  $\bar{\mathbf{d}} \in \mathbb{Z}_+^n$ , and  $\mathbf{y}, \mathbf{z} \in \mathbb{Z}_+^n$  with  $\mathbf{y}, \mathbf{z} \geq \bar{\mathbf{d}}$ . Moreover, let  $V \subseteq \mathbb{Z}^n$  such that  $J_V = J_U : \mathbf{x}^{\bar{\mathbf{d}}}$ . Then

$$\mathbf{y} \leftrightarrow_U \mathbf{z} \iff \mathbf{y} - \bar{\mathbf{d}} \leftrightarrow_V \mathbf{z} - \bar{\mathbf{d}}.$$

*Proof.* This follows immediately using Theorem 4.1: Note that  $\mathbf{y} \leftrightarrow_U \mathbf{z}$  if and only if  $\mathbf{x}^{\mathbf{y}} - \mathbf{x}^{\mathbf{z}} \in J_U$  and similarly,  $\mathbf{y} - \bar{\mathbf{d}} \leftrightarrow_V \mathbf{z} - \bar{\mathbf{d}}$  if and only if  $\mathbf{x}^{\mathbf{y}-\bar{\mathbf{d}}} - \mathbf{x}^{\mathbf{z}-\bar{\mathbf{d}}} \in J_V$ . Now observe that  $\mathbf{x}^{\mathbf{y}-\bar{\mathbf{d}}} - \mathbf{x}^{\mathbf{z}-\bar{\mathbf{d}}} \in J_V$  holds if and only if  $\mathbf{x}^{\mathbf{y}} - \mathbf{x}^{\mathbf{z}} \in J_U$ , as  $J_V = J_U : \mathbf{x}^{\bar{\mathbf{d}}} = \langle \mathbf{x}^{\mathbf{u}} - \mathbf{x}^{\mathbf{v}} : \mathbf{u} + \bar{\mathbf{d}} \leftrightarrow_U \mathbf{v} + \bar{\mathbf{d}} \rangle$ .

Observe that for the backward implication of the equivalence, we employ the condition  $\mathbf{y}, \mathbf{z} \geq \bar{\mathbf{d}}$ , that is,  $\mathbf{x}^{\mathbf{y}-\bar{\mathbf{d}}} - \mathbf{x}^{\mathbf{z}-\bar{\mathbf{d}}}$  is indeed a polynomial in  $\mathbf{x}$ .  $\square$

Note that Theorem 4.2 shows that  $V$  connects any two points  $\mathbf{y}, \mathbf{z}$  in  $\mathcal{C}(\mathbf{y})$  with  $\mathbf{y}, \mathbf{z} \geq \bar{\mathbf{d}}$  via a sequence of points in  $\mathcal{C}(\mathbf{y})$  whose coordinates are also at least as big as  $\bar{\mathbf{d}}$ . Such a path can be constructed by shifting the relation  $\mathbf{y} - \bar{\mathbf{d}} \leftrightarrow_V \mathbf{z} - \bar{\mathbf{d}}$  by  $\bar{\mathbf{d}}$  as shown in Figure 4.1. Moreover, by computing the normal form of  $\mathbf{y}$  with respect to a  $\prec$ -Gröbner basis of  $J_V$  where  $\prec$  is a term ordering that maximizes the  $j$ -th component, we can decide for a given  $\mathbf{d} := \bar{\mathbf{d}} + d_j \mathbf{e}_j$  whether there is some  $\mathbf{z} \in \mathcal{C}(\mathbf{y})$  with  $\mathbf{z} \geq \mathbf{d}$ . This is summarized in Algorithm 1.

**Input** : Let  $U \subseteq \mathbb{Z}^n$ ,  $\mathbf{y}, \mathbf{d} \in \mathbb{Z}_+^n$ . Moreover, let  $S = \text{supp}(\mathbf{d})$ ,  $j \in S$ , and  $\bar{\mathbf{d}} := \mathbf{d} - d_j \mathbf{e}_j$ .  
**Output**:  $\mathbf{z}$  if  $\mathbf{z} \in \mathcal{C}(\mathbf{y})$  with  $\mathbf{z}_j \geq d_j$  exists, False otherwise.

- 1 set  $\bar{\mathbf{d}} := \mathbf{d} - d_j \mathbf{e}_j$ ;
- 2 compute  $V \in \mathbb{Z}^n$  such that  $J_V = J_U : \mathbf{x}^{\bar{\mathbf{d}}} = \langle \mathbf{x}^{\mathbf{u}} - \mathbf{x}^{\mathbf{v}} : \mathbf{u} + \bar{\mathbf{d}} \leftrightarrow_U \mathbf{v} + \bar{\mathbf{d}} \rangle$ ;
- 3 compute a Gröbner basis  $G$  of  $J_V$  w.r.t. a term ordering  $\prec$  that maximizes the  $j$ -th component;
- 4  $\bar{\mathbf{x}}^{\bar{\mathbf{z}}} := \text{NF}_{\prec}(\mathbf{x}^{\mathbf{y}-\bar{\mathbf{d}}}, G)$ ;
- 5 if  $\bar{\mathbf{z}}_j \geq d_j$  then
- 6      $\mathbf{z} := \bar{\mathbf{z}} + \bar{\mathbf{d}} \in \mathcal{C}(\mathbf{y})$ ;
- 7     return  $\mathbf{z}$ ;
- 8     else
- 9         return False;
- 10    end
- 11 end

**Algorithm 1:** Coordinate increment algorithm (CI).

We will now formulate Algorithm 2 that solves Main Problem 2 by iteratively calling the coordinate increment algorithm (Algorithm 1). Algorithm 2 constructs an element  $\mathbf{z} \in \mathcal{C}(\mathbf{y})$  with  $\mathbf{z} + \mathbf{d} \in \mathbb{Z}_+^n$  for a given  $\mathbf{y} \in \mathbb{Z}_+^n$  and  $\mathbf{d} \in \mathbb{Z}^n$  if such an element  $\mathbf{z}$  exists.

**Input** : Let  $U, \mathbf{y}, \mathbf{d}$  be as in Theorem 4.3.  
**Output**:  $\mathbf{z} \in \mathcal{C}(\mathbf{y})$  with  $\mathbf{z} + \mathbf{d} \in \mathbb{Z}_+^n$  if exists, False otherwise.

```

1 set  $S := \text{supp}(\mathbf{d}^-)$ ;
2 let  $\{s_1, \dots, s_k\} := S$  be an enumeration of  $S$  with  $k \leq n$ ;
3 set  $\mathbf{d}^{(1)} := d_{s_1}^- \mathbf{e}_{s_1}$ ;
4 set  $\mathbf{z}^{(0)} := \mathbf{y}$ ;
5 for  $i \in [k]$  do
6   set  $\mathbf{z}^{(i)} := \text{CI}(U, \mathbf{z}^{(i-1)}, \mathbf{d}^{(i)}, s_i)$ ; (with CI from Theorem 4.3)
7   if  $\mathbf{z}^{(i)} == \text{False}$  then
8     return False;
9     STOP;
10  end
11  if  $i < k$  then
12    set  $\mathbf{d}^{(i+1)} := \mathbf{d}^{(i)} + d_{s_{i+1}}^- \mathbf{e}_{s_{i+1}}$ ;
13  end
14 end
15 return  $\mathbf{z}^{(k)}$ ;
```

**Algorithm 2:** Cluster connectivity test (CCT).

**Theorem 4.3.** Let  $U \subseteq \mathbb{Z}^n$ ,  $\mathbf{y} \in \mathbb{Z}_+^n$ , and  $\mathbf{d} \in \mathbb{Z}^n$ . Then Algorithm 2 decides whether there exists  $\mathbf{z} \in \mathcal{C}(\mathbf{y})$  with  $\mathbf{z} + \mathbf{d} \in \mathbb{Z}_+^n$  and if the answer is affirmative it returns  $\mathbf{z}$ .

*Proof.* Let  $S$ ,  $\mathbf{d}^{(i)}$ ,  $\mathbf{z}^{(i)}$ , and  $k$  be as calculated by Algorithm 2. Suppose that  $\mathbf{z} \in \mathcal{C}(\mathbf{y})$  with  $\mathbf{z} + \mathbf{d} \in \mathbb{Z}_+^n$  exists. It suffices to show that at the end of iteration  $i$  we obtain an element  $\mathbf{z}^{(i)} \in \mathcal{C}(\mathbf{y})$  with  $z_{s_j}^{(i)} \geq d_{s_j}^-$  for all  $j \in [i]$ . Then for  $i = k = |\text{supp}(\mathbf{d}^-)|$  the assertion follows as  $z_{s_j}^{(k)} \geq d_{s_j}^-$  for all  $j \in [k]$  and hence  $\mathbf{z}^{(k)} + \mathbf{d} \in \mathbb{Z}_+^n$ .

We have to verify that in iteration  $i$  we can apply Algorithm 1 to the input  $U$ ,  $\mathbf{z}^{(i-1)}$ ,  $\mathbf{d}^{(i)}$ , and  $s_i$ . At the start of iteration  $i$  we have  $\mathbf{z}^{(i-1)} \geq \mathbf{d}^{(i-1)} - d_{s_i}^- \mathbf{e}_{s_i}$ . In the case of  $i = 1$  this is clear as  $\mathbf{d}^{(1)} - d_{s_1}^- \mathbf{e}_{s_1} = \mathbf{0}$  and  $\mathbf{z}^{(0)} = \mathbf{y} \in \mathbb{Z}_+^n$ . For  $i > 1$  this follows by induction as  $\mathbf{z}^{(i-1)} := \text{CI}(U, \mathbf{z}^{(i-2)}, \mathbf{d}^{(i-1)}, s_{i-1})$  with  $\mathbf{z}^{(i-1)} \geq \mathbf{d}^{(i-1)}$  if such an element  $\mathbf{z}^{(i-1)} \in \mathcal{C}(\mathbf{y})$  exists. With  $\mathbf{d}^{(i-1)} = \mathbf{d}^{(i)} - d_{s_i}^- \mathbf{e}_{s_i}$  the claim follows. Thus we can indeed apply Algorithm 1 to the input  $U$ ,  $\mathbf{z}^{(i-1)}$ ,  $\mathbf{d}^{(i)}$ ,  $s_i$  and it returns  $\mathbf{z}^{(i)}$  with  $\mathbf{z}^{(i)} \geq \mathbf{d}^{(i)}$  if such an element  $\mathbf{z}^{(i)} \in \mathcal{C}(\mathbf{y})$  exists, so that the loop is well defined and we obtain  $\mathbf{z}^{(i)} \geq \mathbf{d}^{(i)}$  at the end of each iteration. Let  $i \in [k]$  and observe that  $\mathbf{d}^{(i)} \geq d_{s_j}^-$  for all  $j \in [i]$  so that, together with  $\mathbf{z}^{(i)} \geq \mathbf{d}^{(i)}$ , it follows  $z_{s_j}^{(i)} \geq d_{s_j}^-$  for all  $j \in [i]$ .

In case there is no such  $\mathbf{z} \in \mathcal{C}(\mathbf{y})$  with  $\mathbf{z} + \mathbf{d} \in \mathbb{Z}_+^n$ , there exists  $i \in [k]$  such that  $\text{CI}(U, \mathbf{z}^{(i-1)}, \mathbf{d}^{(i)}, s_i)$  returns ‘False’ and the algorithm stops. This finishes the proof.  $\square$

**Remark 4.4.** Note that in our chemical setting,  $\|\mathbf{d}^-\|_1 \leq 2$ . Therefore, in order to construct some  $\mathbf{z} \in \mathcal{C}(\mathbf{y})$  such that  $\mathbf{z} + \mathbf{d} \in \mathbb{Z}_+^n$ , we have to compute  $J_U : x_i$  only with respect to at most one variable  $x_i$  for each  $\mathbf{d}$ .



Clearly, one can compute suitable Gröbner bases for  $J_U$  and for  $J_U : x_i$  with  $i \in [n]$  in advance and reuse them when constructing the cluster graph. Moreover, note that a generating set of  $J_U : x_i$  can be computed from  $J_U$  via a DegRevLex-Gröbner basis computation with an ordering such that  $x_i$  is minimal (that is, that maximizes the  $i$ -th component), and then removing  $x_i$  from any binomial by division where possible. The resulting generating set of  $J_U : x_i$  is, in fact, still a Gröbner basis of  $J_U : x_i$  that maximizes the  $i$ -th component.

In order to apply Theorem 4.3, we have to compute a Gröbner basis of  $J_U : x_i$  with respect to a DegRevLex term ordering that maximizes some other component  $j$ . As we have already computed a Gröbner basis of  $J_U : x_i$ , we might even use a Gröbner walk to compute the desired DegRevLex-Gröbner basis  $G_{ij}$  of  $J_U : x_i$  that maximizes the  $j$ -th component.

Finally, note that also all the Gröbner bases  $G_{ij}$  for which there is some  $\mathbf{d} \in D$  with  $\text{supp}(\mathbf{d}^-) = \{i, j\}$  can be computed in advance to constructing the cluster graph. The construction of the cluster graph reduces to computing normal forms with respect to these Gröbner bases then.

We will now present an algorithm that reconstructs the cluster graph  $\mathcal{G}(U, D, S)$ .

**Lemma 4.5.** *Let  $\mathbf{s} \in \mathbb{Z}_+^n$  be an initial state and  $M = U \cup -U \cup D \subseteq \mathbb{Z}^n$  be as above. Then Algorithm 3 reconstructs the induced cluster graph  $\mathcal{G}(U, D, S)$ .*

*Proof.* The list  $N$  contains all the nodes that have been discovered but not visited yet. The list  $V$  contains all visited nodes and  $E$  is the edge set that we construct successively. Clearly Algorithm 3 is finite if the cluster graph  $\mathcal{G}(U, D, S)$  is finite: In each iteration of the loop starting in 5 we remove one element from  $N$  and we only add nodes to  $N$  (lines 15/16) if the constructed node is new. It remains to show that the algorithm reconstructs the cluster graph  $\mathcal{G}(U, D, S)$ . We initialize  $N$  with  $\text{NF}(\mathbf{s}, G_U)$  and we successively process all the nodes in  $N$ . For each node  $\mathbf{u} \in N$  we have to decide whether a directed transition  $\mathbf{d}$  can lead to a new adjacent cluster. Therefore we employ Algorithm 2 to construct  $\mathbf{v} := \text{CCT}(U, \mathbf{u}, \mathbf{d})$  if such a  $\mathbf{v}$  exists. In a next step we verify that  $\mathbf{v} + \mathbf{d} \notin \mathcal{C}(\mathbf{u})$ . In this case we found a transition  $\mathbf{d}$  leading to an adjacent cluster and we add the corresponding edge  $(\mathbf{u}, \mathbf{w})$  to  $E$ . Finally we test if  $\mathbf{w} := \text{NF}(\mathbf{v} + \mathbf{d}, G_U) \notin V$ , i.e., we discovered a new cluster and add  $\mathbf{w}$  to  $V$  in this case.  $\square$

We will now explain how the cluster approach can be used to answer Question 1 and Question 2 mentioned in Section 1. Let  $\mathbf{s} \in \mathbb{Z}_+^n$  and  $M = U \cup -U \cup D \subseteq \mathbb{Z}^n$  be as above and let  $\mathcal{G}(U, D, S)$  be the cluster graph computed by Algorithm 3.

**Remark 4.6** (Answer to Question 1). *We can answer Question 1, i.e., given two states  $\mathbf{s}, \mathbf{t} \in \mathbb{Z}_+^n$ , decide whether  $\mathbf{s} \rightarrow_M \mathbf{t}$ . It suffices to calculate, e.g., the shortest path from  $\mathbf{s}$  to  $\mathbf{t}$  in  $\mathcal{G}(U, D, S)$ . If such a path exists, then we have  $\mathbf{s} \rightarrow_M \mathbf{t}$ . Otherwise such a path does not exist.*

As the cluster graph is usually considerably smaller than the state graph, this test can be performed even for larger reaction networks. If one is interested in a specific path from  $\mathbf{s}$  to  $\mathbf{t}$  which, e.g., minimizes the energy needed for the reaction, we can attach the required energy as weights to the edges and then use the shortest path algorithm to compute the desired path. In order to answer Question 2, we have to define exactly when we consider two paths *essentially different*:

**Definition 4.7.** *Let  $\mathbf{s}, \mathbf{t} \in \mathbb{Z}_+^n$  be two states such that  $\mathbf{s} \rightarrow_M \mathbf{t}$ . Further let  $(\mathbf{v}_i)_{i \in [n]}, (\mathbf{w}_i)_{i \in [m]} \subseteq \bar{S}$  be two (directed) paths with  $\mathbf{v}_1 = \mathbf{w}_1 = \mathbf{s}$  and  $\mathbf{v}_n = \mathbf{w}_m = \mathbf{t}$ . Then  $(\mathbf{v}_i)_{i \in [n]}, (\mathbf{w}_i)_{i \in [m]}$  are essentially different if the induced paths in  $\mathcal{G}(U, D, S)$  are different.*

With the definition as above, Question 2 can be answered as follows.

**Remark 4.8** (Answer to Question 2). *Let  $\mathbf{s}, \mathbf{t} \in \mathbb{Z}_+^n$  be two states such that  $\mathbf{s} \rightarrow_M \mathbf{t}$ . Enumerate all paths from  $\mathbf{s}$  to  $\mathbf{t}$  in  $\mathcal{G}(U, D, S)$  by performing a depth-first search on  $\mathcal{G}(U, D, S)$ .*

Usually, enumerating all potential paths in a graph is rather expensive (actually already path counting is #P-complete). The reduction to the cluster graph  $\mathcal{G}(U, D, S)$  reduces the problem size though, so that realistic networks can be tackled.

**Input** : Let  $U, D, \mathbf{s}$  be as in Lemma 4.5.

**Output**: Cluster graph  $\mathcal{G}(U, D, S)$ .

```

1 let  $G_U$  be a Gröbner basis of  $J_U$  w.r.t. to an arbitrarily chosen term ordering;
2 set  $N := \{\text{NF}(\mathbf{s}, G_U)\}$ ;
3 set  $V := \emptyset$ ;
4 set  $E := \emptyset$ ;
5 while  $N \neq \emptyset$  do
6   choose  $\mathbf{u} \in N$ ;
7   set  $N := N \setminus \{\mathbf{u}\}$ ;
8   set  $V := V \cup \{\mathbf{u}\}$ ;
9   for  $\mathbf{d} \in D$  do
10    set  $\mathbf{v} := \text{CCT}(U, \mathbf{u}, \mathbf{d})$ ;
11    if  $\mathbf{v} \neq \text{False}$  then
12      if  $\text{NF}((\mathbf{v} + \mathbf{d}) - \mathbf{u}, G_U) \neq \mathbf{0}$  then
13        set  $\mathbf{w} := \text{NF}(\mathbf{v} + \mathbf{d}, G_U)$ ;
14        set  $E := E \cup \{(\mathbf{u}, \mathbf{w})\}$ ;
15        if  $\mathbf{w} \notin V$  then
16          set  $N := N \cup \{\mathbf{w}\}$ ;
17          set  $V := V \cup \{\mathbf{w}\}$ ;
18        end
19      end
20    end
21  end
22  return  $(V, E)$ ;
23 end

```

**Algorithm 3:** Cluster graph reconstruction (CGR).

## 5. COMPUTATIONAL RESULTS

We will now provide some computational results for Algorithm 3. All computations were performed in CoCoA 4.7 (see [3]) on a machine with a dual core x86\_64 processor with 2Ghz and 2GB of main memory. For the permanganate/oxalic acid reaction and the corresponding mass/charge balance equations (2.1) we obtain 1022 elementary reactions  $M = U \cup -U \cup D$ . We had to construct  $19 \cdot 18 + 1 = 343$  Gröbner bases for the cluster graph reconstruction. All the necessary Gröbner bases computations for this system were performed in less than 2 min (1m53s) and the remaining time for the actual cluster graph reconstruction can be neglected (<1s) as only normal form computations were performed. The Gröbner basis of  $J_U$  contained 136 elements and the Gröbner bases of the corresponding ideals  $J_U : x_i$  were of similar size.

As shown in [9], 16 species do suffice to explain the reaction process. We performed the same computations for the reduced matrix and obtained both, significantly reduced computational time and size of the Gröbner bases. In this case the  $16 \cdot 15 + 1 = 241$  Gröbner bases computations were performed in less than 14 secs and the Gröbner basis of  $J_U$  contained 29 elements. The Gröbner bases of the ideals  $J_U : x_i$  were again of similar size. Again, the time for computing the cluster graph, see Figure 2.1, is neglectable (<1s).

We were also able compute a Gröbner basis of  $J_U$  for a large reaction network from an air pollution model involving 80 species and 13556 reversible elementary reactions. The Gröbner basis of  $J_U$  could be computed in less than 5 min (4m48s) and contained 5740 elements.

## 6. CONCLUDING REMARKS

We have given algorithms to construct the induced subgraph  $\mathcal{G}(U, D, S)$  of  $\mathcal{G}(U, D)$  that is reachable from a given finite set  $S$  of states. From  $\mathcal{G}(U, D, S)$  we can extract useful information about the decomposition of the overall reaction. For example, transitions can be identified that have to occur in any decomposition. Our computations for the permanganate/oxalic acid reaction and the air pollution problem indicate that our algorithms are applicable in practice even for real-world size problems with a large number of species. Therefore, we are confident that our cluster network approach will turn out very useful in the study of more complex biochemical reaction systems.

## REFERENCES

- [1] E. Abel. Über den Verlauf der Reaktion zwischen Wasserstoffsperoxyd und salpetriger Säure. *Monatshefte für Chemie/Chemical Monthly*, 83(5):1111–1115, 1952.
- [2] S.J. Adler and R.M. Noyes. The Mechanism of the Permanganate-Oxalate Reaction. *Journal of the American Chemical Society*, 77(8):2036–2042, 1955.
- [3] CoCoATeam. CoCoA: a system for doing Computations in Commutative Algebra. Available at <http://cocoa.dima.unige.it>.
- [4] D.A. Cox, J.B. Little, and D. O’Shea. *Ideals, varieties, and algorithms: an introduction to computational algebraic geometry and commutative algebra*. Springer Verlag, 2007.
- [5] E. Deiss. Fachgruppensitzungen von Donnerstag, den 27. bis Sonnabend, den 29. Mai. *Zeitschrift für Angewandte Chemie*, 39:664–665, 1926.
- [6] P. Diaconis and B. Sturmfels. Algebraic algorithms for sampling from conditional distributions. *Annals of Statistics*, pages 363–397, 1998.
- [7] J.-L. Faulon and A.G. Sault. Stochastic generator of chemical structure. 3. reaction network generation. *Journal of Chemical Information and Computer Sciences*, 41(4):894–908, 2001.
- [8] A.V. Harcourt and W. Esson. On the laws of connexion between the conditions of a chemical change and its amount. *Philosophical Transactions of the Royal Society of London*, 156:193–221, 1866.
- [9] U.U. Haus and R. Hemmecke. Decomposition of reaction networks: Unraveling the initial phase of permanganate/oxalic acid reaction. *submitted*, 2009.
- [10] R. Hemmecke and P.N. Malkin. Computing generating sets of lattice ideals and markov bases of lattices. *Journal of Symbolic Computation*, In Press, Accepted Manuscript:–, 2009.
- [11] K. Kovács, B. Vizvári, M. Riedel, and J. Tóth. Decomposition of the permanganate/oxalic acid overall reaction to elementary steps based on integer programming theory. *Physical Chemistry Chemical Physics*, 6(6):1236–1242, 2004.
- [12] K.A. Kovacs, P. Grof, L. Burai, and M. Riedel. Revising the mechanism of the permanganate/oxalate reaction. *Journal of Physical Chemistry A*, 108(50):11026–11031, 2004.
- [13] M. Kreuzer and L. Robbiano. *Computational commutative algebra 1*. Springer, 2000.
- [14] W. Marwan, A. Wagler, and R. Weismantel. A mathematical approach to solve the network reconstruction problem. *Math. Methods of Operations Research*, 67:117–132, 2008.
- [15] E.W. Mayr and A.R. Meyer. The complexity of the word problems for commutative semigroups and polynomial ideals. *Adv. Math*, 46(3):305–329, 1982.
- [16] V. Pimienta, D. Lavabre, G. Levy, and JC Micheau. Reactivity of the Mn (III) and Mn (IV) Intermediates in the Permanganate/Oxalic Acid/Sulfuric Acid Reaction: Kinetic Determination of the Reducing Species. *The Journal of Physical Chemistry*, 98(50):13294–13299, 1994.
- [17] A. Shiu and B. Sturmfels. Siphons in chemical reaction networks. *preprint arXiv:0904.4529*, 2009.
- [18] A. Skrabal. Über die Primäroxidtheorie der Oxydationsprozesse. *Zeitschrift für anorganische Chemie*, 42(1), 1904.
- [19] I. Szalkai. A new general algorithmic method in reaction syntheses using linear algebra. *Journal of Mathematical Chemistry*, 28(1):1–34, 2000.

UNIVERSITY OF MAGDEBURG, GERMANY  
E-mail address: haus@imo.math.uni-magdeburg.de

UNIVERSITY OF MAGDEBURG, GERMANY  
Current address: Technische Universität Darmstadt, Germany  
E-mail address: hemmecke@mathematik.tu-darmstadt.de

TECHNISCHE UNIVERSITÄT DARMSTADT, GERMANY  
*E-mail address:* pokutta@mathematik.tu-darmstadt.de